الجمهورية الجزائرية الديمقراطية الشعبية
People`s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
National School of Cyber Security
Foundation Training Department
وزارة التعليم العالي والبحث العلمي
المدرسة الوطنية العليا في الأمن السيبراني
قسم التكوين القاعدي

Graph Theory answer key

Exercise 1: (04 points)

Select all correct answers (multiple correct answers possible). +0.5 point for each correct answer -0.25 point for each incorrect one.

1. In an undirected graph, the sum of the degrees of all vertices is equal to:
   a) The total number of edges
   b) **Twice the number of edges**
   c) The square of the number of vertices
   d) The total number of vertices

2. A bipartite graph is a graph in which:
   a) **Edges connect two disjoint vertex sets**.
   b) Every vertex has an even degree
   c) The graph is directed
   d) **There is no cycle of odd length**

3. The chromatic number of a graph is always:
   a) **Less than or equal to the number of vertices**
   b) Equal to the maximum degree
   c) **Greater than or equal to the size of the largest clique**
   d) Even if the graph is planar

4. If a planar graph has 6 vertices and 10 edges, how many faces does it have?
   a) 4
   b) **5**
   c) 6
   d) 8

5. In a directed graph, the sum of all out-degrees equals:
   a) Zero
   b) The total number of vertices
   c) **The total number of edges**
   d) Twice the number of edges

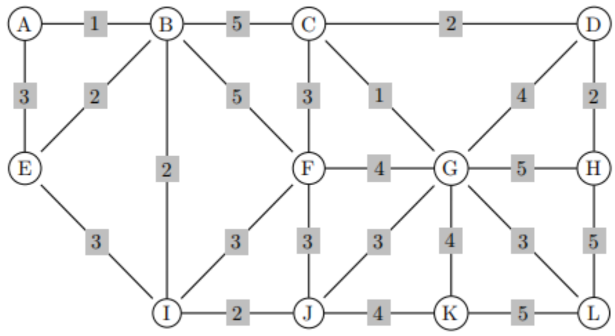
6. In a flow network, a valid flow must satisfy:
   a) **The conservation law at each vertex (except source the sink)**
   b) **That the flow on each edge does not exceed its capacity**
   c) That the flow is symmetric across arcs
   d) That the total flow equals the sum of the capacities

7. The Bellman-Ford algorithm:
   a) **Can handle negative weights**
   b) Does not work if the graph is undirected
   c) **Can detect a negative-weight cycle**
   d) Is always faster than Dijkstra's algorithm

8. A cut in a flow graph:
   a) **Is a set of edges that separates the source from the sink**
   b) **Can be used to limit the flow**
   c) Has a capacity equal to the sum of the flows on its edges
   d) Is always unique

Exercise 2: (05 pts)

We want to build a network with a minimum cost to connect 12 switches. The wiring costs are given by the graph below.



Following a political decision, the wired connections G-H and A-E are imposed (the network must include G-H and A-E). To determine the minimum cost wiring that respects these constraints, you must answer the following questions:

1. Which algorithm should you use to determine the minimum cost wiring?

   To determine the minimum cost, we must use the algorithm dedicated to the spaning tree. In this case, we use either Kruskal or Prim Algorithm. The most adequate for this exercise is Kruskal Algorithm. (1.5 pts)

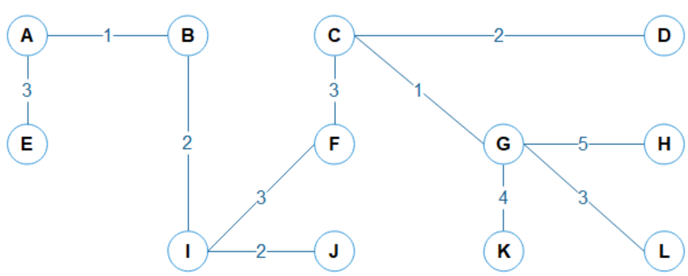
2. How should this algorithm be adapted to the specific case of the exercise, considering the imposed constraints (the connections G-H and A-E must be part of the network)?

   The Kruskal Algorithm will be adapted by inserting the two connections G-H and A-E at the top of the ordered list of edges. (1 pts)

3. Apply the algorithm step by step to obtain the minimum wiring, respecting the constraints, and specify the total cost.

1st Step: order the edges in ascending order, and then move the edges E-A and G-H at the top of this order. So the order of edges will be as follow: (1 pts)

E-A
G-H
A-B
C-G
E-B (Reject)
B-I
I-J
C-D
D-H (Reject)
E-I (Reject)
I-F
C-F
F-J (Reject)
J-G (Reject)
G-L
F-G (Reject)



G-K
K-J (Reject)
G-D (Reject)
B-C (Reject)
B-F (Reject)
H-L (Reject)
L-K (Reject)

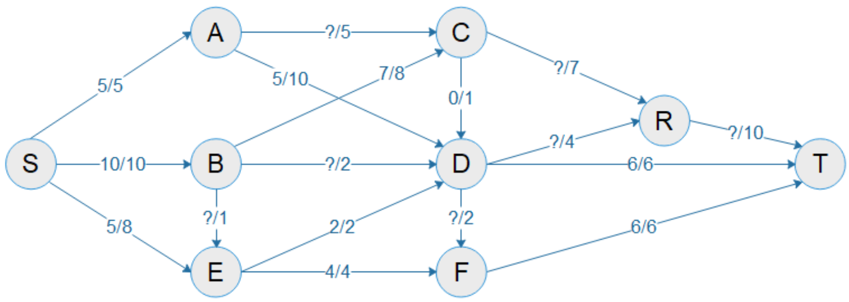
2nd Step: We select, in this order, edges which do not form a cycle. So, the selected edges are: E-A, G-H, A-B, C-G, B-I, I-J, C-D, I-F, C-F, G-L, G-K. (1 pts)

The obtained spaining tree that respect the constrains is showed above.

The total cost is equal to 29 (0.5 pts)

Exercise 3: (05pts)

Before starting a highway construction project, a company is evaluating the capacity of the existing road network, represented by the graph below, which connects city **S** to city **T**. To do so, it estimated the maximum number of vehicles that can travel along each road per hour. These capacities are given in hundreds of vehicles per hour. For example, the arc from **S** to **E** has a capacity of 8 (i.e., 800 vehicles/hour) and a current flow of 5 (i.e., 500 vehicles/hour), which is noted as 5/8 on the arc.



1. Complete the initial flow indicated on the graph. What is the total value of this flow?

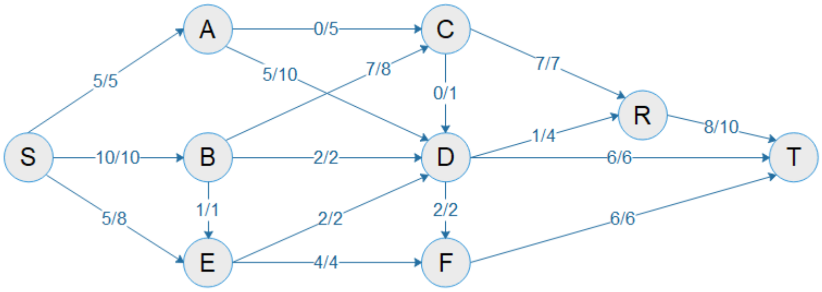
Edge	**A-C**	**B-E**	**B-D**	**C-R**	**D-F**	**D-R**	**R-T**
flow	**0** (0.2 pts)	**1**(0.25 pts)	**2**(0.25 pts)	**7**(0.2 pts)	**2**(0.2 pts)	**1**(0.2 pts)	**8**(0.2 pts)

The total value of this flow is equal to 20 (0.5 pts)

2. Is this flow maximal? Justify your answer. If not, use this initial flow as a starting point and apply the Ford-Fulkerson algorithm to compute the maximum possible flow of vehicles from S to T.

No, this flow is not maximal. (0.25 pts)

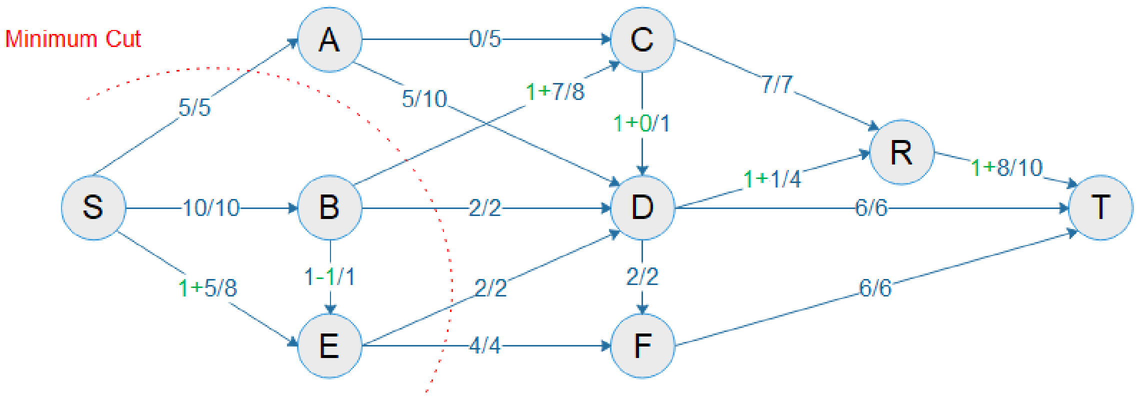
The initial flow is below



Steps of the Ford-Fulkerson algorithm

step	**Chain**	**γ1**	**γ2**	**γ**
1	**S-E-B-C-D-R-T**	**2**	**1**	**1**

After the first step the new flow is below, and it is the maximum. (1.25 pts)

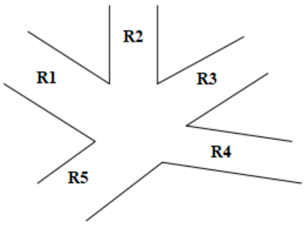


3. Determine and draw a minimal cut of the network. Compare the capacity of this cut to the value of the maximum flow. What do you conclude?

The minimum cut is represented in the figure above. The edges that compose this cut are: S-A, B-C, B-D, E-D , and E-F (0.5 pts). The capacity sum of these edges is equal to 21 and the maximum flow is equal to 21 (0.5 pts). So, we conclude that the maximum flow must equal to the minimum cut (0.5 pts).

Exercise 4: (06 pts)

An intersection is represented by the figure below. The possible movements through this intersection (referred to as ***crossings***) are listed in the accompanying table. For example, according to this figure, the movements (***crossings***) R1–R3 and R2–R5 cannot be allowed simultaneously, and other combinations of movements present similar conflicts.



Going from	R1	R2	R3	R4	R5
to....	R3, R5	R1, R5, R4	R1, R4	R3, R1	R3, R4

We can represent this situation with a graph in which, each vertex (***crossing***) represents a possible movement through the intersection (e.g., going from R1 to R3), and an edge between two **crossings** indicates that the corresponding crossings are incompatible and cannot occur at the same time (for safety or collision reasons).

1. List all pairs of incompatible crossing (0.05 pts for each pair of crossing conflicts)

(R1-R3) conflicts with (R4-R3), (R5-R3), (R2-R5), (R2-R4), (R4-R1)

(R1-R5) conflicts with (R2-R5),

(R2-R1) conflicts with (R3-R1), (R4-R1)

(R2-R5) conflicts with (R1-R3), (R1-R5), (R3-R1), (R4-R1)

(R2-R4) conflicts with (R1-R3), (R3-R1), (R3-R4), (R4-R1), (R5-R3), (R5-R4)

(R3-R1) conflicts with (R2-R1), (R2-R5), (R2-R4), (R4-R1)

(R3-R4) conflicts with (R1-R3), (R2-R4), (R4-R1), (R5-R3), (R5-R4)

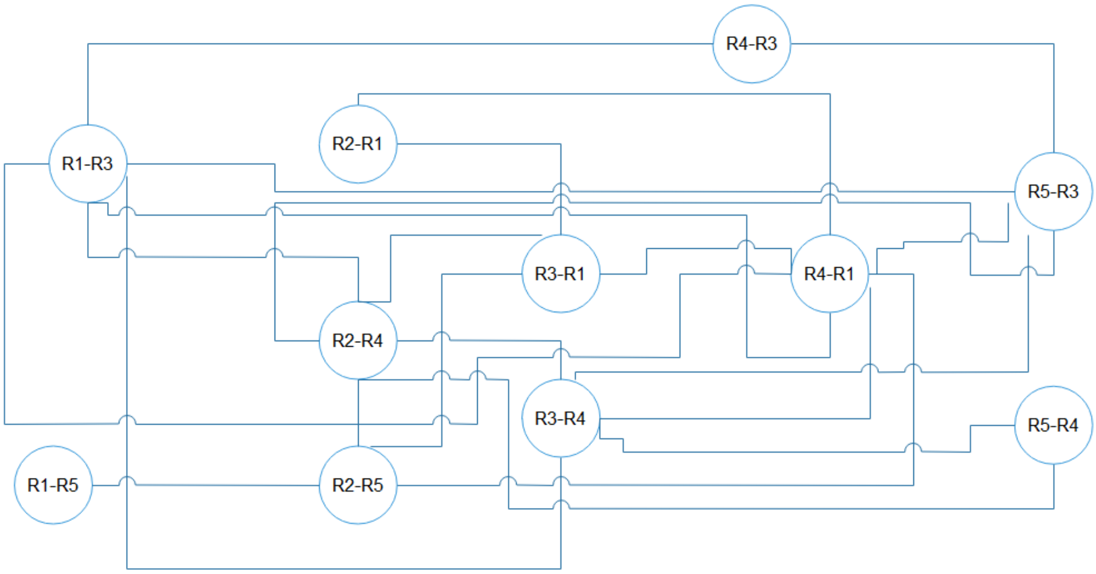
(R4-R3) conflicts with (R1-R3), (R5-R3)

(R4-R1) conflicts with (R1-R3), (R2-R4), (R3-R1), (R3-R4), (R5-R3)

(R5-R3) conflicts with (R1-R3), (R2-R4), (R3-R4), (R4-R3), (R4-R1)

(R5-R4) conflicts with (R2-R4), (R3-R4)

2. Model these incompatibilities using a graph. (1pts)



3. Apply the Welsh-Powell algorithm to color the obtained graph.

1st Step:

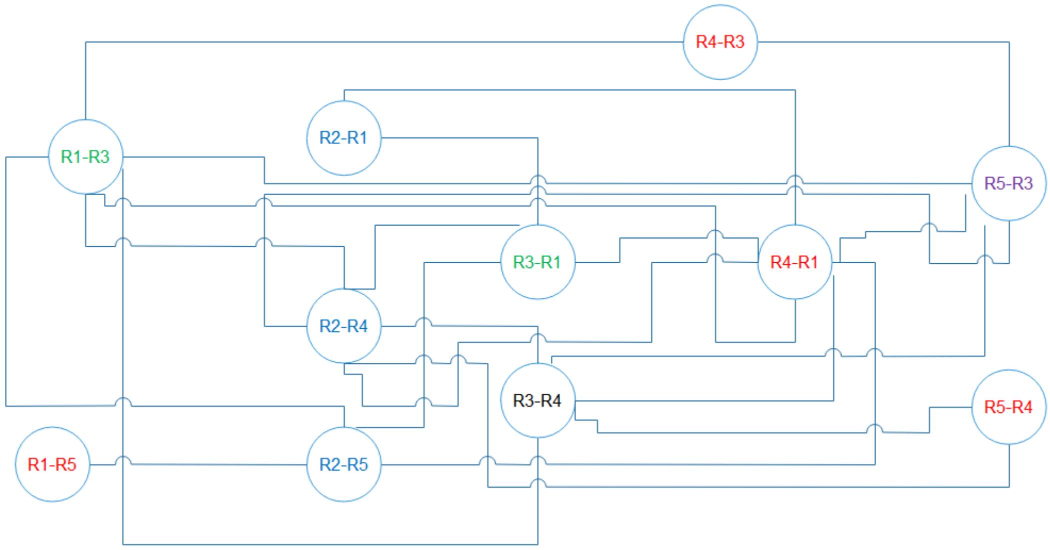
Classify vertices according to their degree (0.5pts)

Vertex	R4-R1	R1-R3	R2-R4	R5-R3	R3-R4	R2-R5	R3-R1	R2-R1	R4-R3	R5-R4	R1-R5
degree	7	6	6	5	5	4	4	2	2	2	1

2nd Step:

Assign colors numbered from 1 onward, and for each vertex, assign the smallest available color that has not been used by its adjacent vertices. (0.5pts)

Vertex	R4-R1	R1-R3	R2-R4	R5-R3	R3-R4	R2-R5	R3-R1	R2-R1	R4-R3	R5-R4	R1-R5
color	C1	C2	C3	C5	C4	C3	C2	C3	C1	C1	C1



4. Explain what a set of vertices having the same color represents. (1pts)

Vertices assigned the same color are not adjacent, meaning there is no conflict between them

5. Explain what the chromatic number of the graph means in terms of managing traffic at the intersection. (1pts)

The chromatic number indicates that there are five groups of vehicles that cannot move at the same time, but each group sharing the same color can be scheduled to move simultaneously without any issues.

Groupe	**Crossing without any issues**
C1	R1-R5, R4-R3, R4-R1, R5-R4
C2	R1-R3, R3-R1
C3	R2-R1, R2-R5, R2-R4
C4	R3-R4
C5	R3-R3